Teaching a new dog old tricks – the Assumption Architecture

Jakob Fredslund

We present a learning system intended for the simplistic ecological niche of LEGO robots. Binary feedback in the form of 'good!' and 'bad!' from a trainer is enough for the robot to learn a variety of simple tasks. The robot's sensors are structured in groups called 'senses' that pre-process sensor data into useful information. The robot forms consistent assumptions about when to take what action. All feedback is considered significant and may have great impact on the learned assumptions; this is one of the reasons for fast learning. Many experiments were done in simulation, one also with a real LEGO robot.

I. Introduction

What tasks could a simple LEGO robot ever do? Equipped with only switch, light, temperature, and angle sensors it would never be able to do face recognition, no matter how its control program was organised. Still, there are a number of simple, yet non-trivial, tasks that lie within the limits set by the sensors, such as light following, obstacle detection, line following, etc. In fact, you might say that the tasks that a LEGO robot is capable of performing are tasks you would require as basic skills of a more sophisticated robot. Such basic skills might form the lower layers of a Behaviour Based control program designed for more complex tasks [3]. Hence, they could be considered fundamental: if the robot can't do this, it can't do anything.

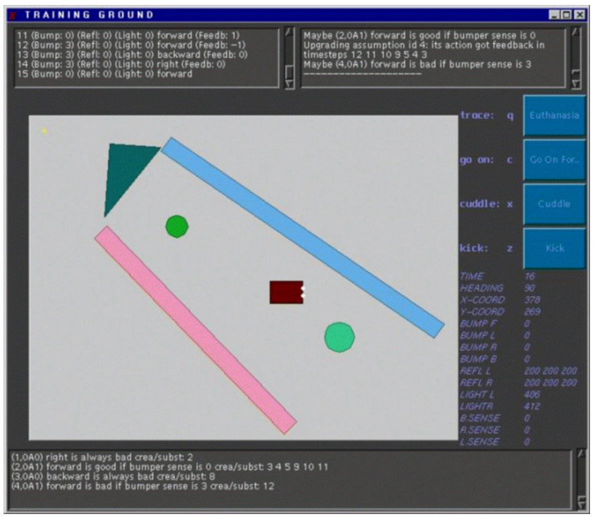
What can you teach a robot by saying only *good*! and *bad*! to it? This sort of simple, binary training is immediately comprehensible, even for children, and the trainer needs no prerequisites. The principle is incorporated in all learning methodologies in some form since learning is about choosing one behaviour instead of another on the grounds of some criterion, which corresponds to classifying one behaviour as good and others as bad, in the current situation. With only this core element and without all other forms of helpful trainer feedback, the resulting learning scheme is a kind of 'back to basics', following the Animat Approach: How little is necessary [12]. With less than this, learning is not possible.

Can a LEGO robot be taught these basic skills through binary training? In this work, we present a rule-based learning system for a simulated LEGO robot. During binary training the robot forms assumptions about when to do what, and over time these assumptions converge more and more into rules. We show that it is possible to successfully transfer the learned rules to a physical LEGO robot. The system has been named the Assumption Architecture.

LEGO-Lab, Dept. of Comp. Science, Aarhus University, Denmark.

II. The Assumption Architecture

In our learning system, a virtual LEGO robot, Mr. Mind, inhabits a simulator (cf. fig.1). A trainer teaches him various simple tasks by giving either ☺ (*good*!), ☹ (*bad*!), or 😐 (*go on*). To keep things synchronised, having taken an action Mr. Mind will wait for trainer feedback before taking the next action. The *history* holds all previous sensor readings, actions taken and received trainer feedback.



```
11 (Bump: 0) (Refl: 0) (Light: 0) forward (Feedb: 1)
12 (Bump: 3) (Refl: 0) (Light: 0) forward (Feedb: −1)
13 (Bump: 3) (Refl: 0) (Light: 0) backward (Feedb: 0)
14 (Bump: 3) (Refl: 0) (Light: 0) right (Feedb: 0)
15 (Bump: 0) (Refl: 0) (Light: 0) forward
```

```
Maybe (2,0A1) forward is good if bumper sense is 0
Upgrading assumption id 4: its action got feedback in
timesteps 12 11 10 9 5 4 3
Maybe (4,0A1) forward is bad if bumper sense is 3
-----------------------
```

```
TIME       16
HEADING    90
X-COORD    378
Y-COORD    269
BUMP F     0
BUMP L     0
BUMP R     0
BUMP B     0
REFL L     200 200 200
REFL R     200 200 200
LIGHT L    406
LIGHTR     412
B.SENSE    0
R.SENSE    0
L.SENSE    0
```

```
(1,0A0) right is always bad crea/subst: 2
(2,0A1) forward is good if bumper sense is 0 crea/subst: 3 4 5 9 10 11
(3,0A0) backward is always bad crea/subst: 8
(4,0A1) forward is bad if bumper sense is 3 crea/subst: 12
```

Figure 1: The simulator.

A. Senses

"Having a camera does not mean that you have vision." [8]. A camera gives you *data*; what you want is *information*. In order to get information out of data, you need to interpret the *quantitative* values into *qualitative* values. Humans do it all the time: consciously we are not bothered with which perceptrons in our eyes that fire, and which that do not; eventually our brain is just rendered the information "this is dad".

”Perception [is] knowing what in the environment is relevantly the case” [12]. *Relevant* means with respect to some objective, and if we know that the two light reflection sensors underneath the robot will be used for detecting lines, then the absolute values of the two sensors are unimportant. The absolute values are data; the information we want is whether or not there is a line. On these two levels of abstraction the search spaces are $V \times V$ versus $\{$ *edge*, *left boundary of edge*, *right boundary of edge, no edge* $\}$, respectively, where V is the set of possible values of the sensor. Doing this abstraction in a pre-processing sensor module, instead of forcing the learning algorithm itself to do it, results in a huge reduction of the search space. The learning can be based on a much more precise and succinct sensor signal

The sensors of Mr. Mind can naturally be divided into three groups: four bumper sensors (one on each side), two reflection sensors (underneath), and two ambient light sensors. Within each group, the sensors measure the same aspect of the environment, analogously to the human senses that also use several sensors: e.g. two eyes for the vision sense. A 'wrapper' is attached to each group of sensors; the wrapper is a small module that pre-processes the raw input sensor data and outputs relevant information in the form of a single integer value. A group of sensors and the associated wrapper together constitute a *sense.*

This modularity increases learning speed due to the reduction of the search space, and it is also a way of achieving decentralisation. (In [6] a similar, modular and decentralised way of structuring sensor input is presented). Further, an easier transfer from simulation to reality might be expected; the senses can be made robust by letting them focus on *difference* or change of sensor values rather than absolute values. Thus, it does not matter that the reflection sensors show (23, 81) in the simulator and (69, 148) in the real world if the value returned by the reflection sense is simply +1 (for *right boundary of edge*) when the right sensor value is significantly greater than the left one. Also, this makes the simulator much simpler since there is no need to model real world values very accurately.

Mr. Mind has four actions: **forward**, **backward, left**, **right**. Each is a wrapper around low level motor commands; e.g. **left** is to go forward on one motor and backward on the other with certain fixed time and speed parameters. Thus we can talk about perception and action rather than of sensor values and motor commands – i.e. on a higher level of abstraction.

B. Assumptions

Say that the trainer keeps giving 😐 for a while (being ‘neutral’), and then the first positive feedback ☺ is given after a **right**. There might be dozens of reasons for the ☺. Mr. Mind has no chance of knowing the reason before he has more experience to judge from, so he assumes for the time being that **right** is *always* good. This is probably not true, but, for all he knows, it could be. And indeed, if the trainer never gives a ☹ following a **right**, it might actually be the case that **right** is always good and the task is “turn on the spot”. Thus, the *assumption* ”**right** is always good” is created. All assumptions are on the form

$\langle action \rangle$ is good/bad if $\langle condition \rangle$

The assumptions are formed from a list of applicable *assumption types* ordered by simplicity, the simplest one being the ‘always’-one (where the condition is just TRUE). The given list of assumption types defines explicitly the kinds of stimuli/response-connections Mr. Mind can comprehend. The assumption types are divided into classes. Class 0 (referred to as 0A) consists of those types whose conditions involve no more than the current time step. The ‘good’ ones are (the ‘bad’ ones are analogous):

0A0: $\langle action \rangle$ is always good

0A1: $\langle action \rangle$ is good if $\langle sense\ condition \rangle$

0A2: $\langle a \rangle$ is good if $\langle sc_1 \rangle$ AND $\langle sc_2 \rangle$

0A3: $\langle a \rangle$ is good if $\langle sc_1 \rangle$ AND $\langle sc_2 \rangle$ AND $\langle sc_3 \rangle$

(‘a’ is short for action, ‘sc’ for sense condition, i.e., a condition on one of the senses, like “light sense is 3”).

These types correspond to direct sensory-motor connections. The behaviour of the simplest Braitenberg-creatures [2] could be expressed with assumptions of class 0A. The next class, 1A, consists of those types that involve the action taken in the previous time step, e.g.

1A0: $\langle a_1 \rangle$ is good if the previous action was $\langle a_2 \rangle$

Similarly, class 2A is the class of types involving the action taken two time steps before, e.g.:

2A0: $\langle a_1 \rangle$ is good if second-to-last action was $\langle a_2 \rangle$

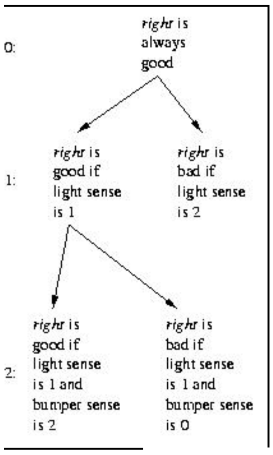
Currently, no more than these assumption types have been implemented; of course the <*sense condition*>-types (0A1, 0A2, and 0A3) could be adapted to fit in classes 1A and 2A as well; for example, one could imagine an assumption like

$\langle action \rangle$ is good if $\langle sense\ condition \rangle$ was true in the previous time step

Now, whenever a new, *live* assumption is instantiated from one of these types, it is associated with a unique ID and the time step where it was created, its *creator*.

If, in later time steps, further trainer feedback supports the assumption, these time steps are also associated with it as *substantiators*, much like in Truth Maintenance Systems where each item of knowledge has a list of other items that support belief in it.

Now imagine that the task is not "turn on the spot" but instead "follow the light", and that the first **right** was good since at that time the light source was to the right of Mr. Mind. Now, however, it is to the left, and therefore the next **right** is followed by ☹. Since **right** can no longer be assumed "always good", the assumption "**right** is always good" is *refuted* and new ones have to be formed that are consistent with respect to both the ☺, and the ☹.



0:
right is always good
1:
right is good if light sense is 1
right is bad if light sense is 2
2:
right is good if light sense is 1 and bumper sense is 2
right is bad if light sense is 1 and bumper sense is 0

Figure 2: The refining of an assumption. The leaves of the trees are the live assumptions; their types are shown to the left. The light sense value 1 means "more light to the right", 2 is "more light to the left".

In fact, the assumption is not discarded, rather it is *refined*: it turned out to be too general, but still, apparently there *are* cases where **right** is good, and a new assumption about the nature of these cases is then formed, involving the light sense. Additionally, an assumption about the cases where **right** is *not* good is formed. This is illustrated in figure 2. More feedback might result in further refining of assumptions. When an assumption is refuted, it will be replaced by one of higher type, i.e. a more specialised one. Much like in the game of Master Mind, the robot tries to find the hidden pattern in the trainer's feedback; hence the name Mister Mind.

C. *Algorithm*

A principle of *significance of all feedback* is adopted in order to fully exploit the trainer feedback (since Mr. Mind takes one action and then waits for the trainer to react with ☺, ☹, or 😐 before taking the next action, the trainer does not risk 'being late' with the feedback accidentally rewarding the wrong action). After each non-neutral feedback, several things can happen: either (1) a live assumption is substantiated, or (2) a new assumption is formed. Further, (3) a live assumption can be refuted and subsequently refined. All ☺ and ☹ are considered significant in the sense that, since no inconsistencies are allowed, a single ☹ can have substantial impact on the live assumptions. In other words, the trainer is taken very seriously. The learning system relies on these two invariants:

- Inv1: After each time step, the live assumptions are consistent with the history.
- Inv2: Each time step where ☺ or ☹ occurred is represented exactly once in an assumption as either a creator or a substantiator.

This is an example of an assumption from an actual run (the parenthesised prefix is its ID and type):

(2, 0A1) **right** is good if bumper sense is 3 (creator/substantiators: 32 41 46)

This assumption, call it A_2, was created in time step 32 following a ☺. In time steps 41 and 46, the bumper sense was also 3 (i.e., the condition of A_2 was true), Mr. Mind took a **right**, and the trainer gave ☺, exactly as in time step 32 (and as A_2 predicted). Assumptions whose conditions are true are said to be *fulfilled*; thus A_2 was fulfilled in time steps 41 and 46, and their ☺'s substantiated A_2. By Inv2, the time steps 32, 41, and 46 are not associated with any other assumption.

Now, when A_2 was created in time step 32, first the "always" type, 0A0, was tried, but that somehow proved inconsistent with the history. Hence, it was refined to one of type 0A1. When an 0A1-assumption is created, the bumper sense, the reflection sense, and the light sense are tried successively as the *triggering* sense. In this example, no evidence was found that the bumper sense value 3 of time step 32 was not the reason for the ☺, and so the bumper sense was used with the value 3 to create the assumption "**right** is good if bumper sense is 3".

Say that in time step 54, the bumper sense is again 3 and Mr. Mind takes a **right**. And then suddenly a ☹ results. Now A_2 is refuted and to keep with Inv1, its creator and substantiators will have to be re-treated. The creator, 32, may give rise to a refined assumption of type 0A2, but the former substantiatiors do not necessarily substantiate this new version of A_2. Imagine that incidentally in both time steps 41 and 46, the light sense was 1 at the same time as the bumper sense being 3 (cf. fig. 3). Imagine that the trainer did in fact *not* reward the **right**-actions in time steps 41 and 46 because of the bumper sense being 3, but because the light sense was 1. Since A_2 was created

time step	bumper sense	reflection sense	light sense	action	trainer feedback
32	3	1	1	**right**	☺
41	3	0	1	**right**	☺
46	3	0	1	**right**	☺
54	3	0	2	**right**	☹

Figure 3: Excerpt of an example history.

in time step 32 already, the ☺'s of time steps 41 and 46 were then regarded as substantiating A_2, since the bumper sense was in fact 3, even though the ☺'s were given for another reason. This reason is not discovered until time step 54 when A_2 is refuted. Then, 41 gives rise to a new assumption: An 0A1-assumption is tried, first with the bumper sense as the triggering sense, but this proves inconsistent because of the ☹ of time step 54. Next, the reflection sense is up, but this also fails due to time step 54. At last the light sense is tried, and this works out since the light sense was 1 in time step 41 but 2 in 54. Thus, the new assumption A_3, "**right** is good if light sense is 1", is created, and subsequently 46 is recognised as substantiating it. Finally, A_4, "**right** is bad if light sense is 2", is formed with creator 54.

When an assumption is refuted, Mr. Mind will not take the next action until its creator and substantiators have been treated, thus restoring Inv1. If it turns out that no new assumption can be consistently created, the trainer has given inconsistent feedback. Because of the principle of significance of all feedback a training mistake is a potentially serious situation. Either the trainer made a plain mistake, or (s)he wants Mr. Mind to learn a quibbling connection between sense values and action that cannot be expressed by any of the given assumption types – a case of 'ethnocentric oversight' where the human forgets about the capabilities of the robot. From Mr. Mind's point of view, there is no way of telling the difference.

Either way, something has to be done. One possibility would be to *forget*: to pretend it never happened and go on leaving an inconsistency behind, but that would violate Inv1. Instead, Mr. Mind will *forgive* his trainer and ignore the feedback of one time step to restore consistency. Say that A_4 from the above example is refuted in time step 74, and that its creator 54 fails to give rise to a new assumption. Ignoring (the feedback of) time step 54 will clear things up: A_4 is deleted and Inv1 is restored. In this case, however, Mr. Mind is forced to ask himself an existential question: "When this was wrong, why should everything else I believe in be right?" The supposedly erroneous feedback of time step 54 influenced the refutation of other assumptions. Therefore a whole new set of assumptions is created from scratch by going through all feedback in history.

If, after time step 54, A_4 had been substantiated by several more time steps that also could not give rise to new assumptions (or substantiate existing ones) after the refutation of A_4 all these time steps would have to be ignored to restore consistency. In other words, the trainer must then have made several mistakes rather than just one. There is no way of knowing whether this is actually the case, but due to Inv1 there was consistency immediately before the latest time step 74, and so ignoring just 74 will also restore consistency. And this is what is done: if only one earlier time step cannot be re-treated, ignore it. If more than one cause problems, ignore instead the latest time step.

After analysing the trainer feedback, Mr. Mind assigns weights to the actions and picks one probabilistically. All fulfilled assumptions can be 'put to the test': if, e.g., the light sense is 1 then the assumption A_3, "**right** is good if light sense is 1", is fulfilled. Doing a **right** would therefore leave the trainer with the chance of refuting or substantiating further A_3. This is reflected by increasing the weight of **right** by the total number of time steps supporting it. That is, the more it previously proved a good idea, the higher the chance of doing it again. For fulfilled "bad"-assumptions, weight is subtracted.

A *conflict* is a situation where two fulfilled assumptions concerning the same action predict opposite trainer feedback. This does not mean inconsistency with the history; it just means that the conditions of the two assumptions have never before been true simultaneously. For example, the two assumptions "**right** is good if bumper sense is 3" and "**right** is bad if light sense is 1" could very well temporarily co-exist. If a situation arises where the bumper sense is 3 and the light sense is 1, these assumptions cannot both be correct. Such situations are *interesting* [8] in that Mr. Mind is sure to increase his knowledge if (1) he takes the action concerned (and so he does), and (2) the trainer gives feedback, because then one of the assumptions will be refuted and subsequently refined, while the other will be substantiated. Further, the trainer has the possibility of positioning Mr. Mind on the interesting places on the arena in order to let him discover the essence of the task more quickly; e.g. by placing him close to the line when the task is line following.

Given that the trainer is trying to teach Mr. Mind something that can be expressed with the list of assumption types, given that there is no simpler explanation that is consistent with history, and given that no inconsistent training occurs, then Mr. Mind will eventually find the right assumptions that account for the trainer's feedback: due to the invariants, at all times the set of live assumptions can account for the trainer feedback seen so far. To refine this set into a final set of assumptions, Mr. Mind must experience the cases that distinguish this set from the final set. If the trainer gives correct feedback in these cases, a set of assumptions that solve the task will be found.

D. *Experiments*

The generality of the assumption types was demonstrated in a number of different experiments, one of which was also done with a physical robot. The first experiment was simple line following (cf. fig. 4). Within 40 time steps, Mr. Mind had learned several assumptions, including:

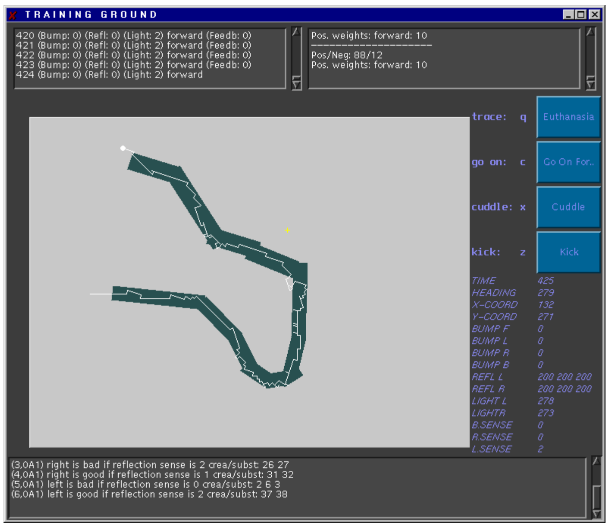
(0, 0A1) **forward** is good if reflection sense is 0
(4, 0A1) **right** is good if reflection sense is 1

(6, 0A1) **left** is good if reflection sense is 2

(where the values of 0, 1, and 2 mean “*no edge*”, “*edge; dark side to the right*”, and “*edge; dark side to the left*”, respectively), corresponding to the rules that were also used in [10]:

1) If you see (x, x) go forward
2) If you see (bright, dark) turn right
3) If you see (dark, bright) turn left

Recall that initially there is no indication anywhere of the task that Mr. Mind is supposed to learn. These assumptions were the simplest ones that matched the feedback – and the reflection sense turned out to be the triggering sense in this task.



```
420 (Bump: 0) (Refl: 0) (Light: 2) forward (Feedb: 0)
421 (Bump: 0) (Refl: 0) (Light: 2) forward (Feedb: 0)
422 (Bump: 0) (Refl: 0) (Light: 2) forward (Feedb: 0)
423 (Bump: 0) (Refl: 0) (Light: 2) forward (Feedb: 0)
424 (Bump: 0) (Refl: 0) (Light: 2) forward
```

```
Pos. weights: forward: 10
----------------------
Pos/Neg: 88/12
Pos. weights: forward: 10
```

```
TIME       425
HEADING    279
X-COORD    132
Y-COORD    271
BUMP F     0
BUMP L     0
BUMP R     0
BUMP B     0
REFL L     200 200 200
REFL R     200 200 200
LIGHT L    278
LIGHTR     273
B.SENSE    0
R.SENSE    0
L.SENSE    2
```

```
(3,0A1) right is bad if reflection sense is 2 crea/subst: 26 27
(4,0A1) right is good if reflection sense is 1 crea/subst: 31 32
(5,0A1) left is bad if reflection sense is 0 crea/subst: 2 6 3
(6,0A1) left is good if reflection sense is 2 crea/subst: 37 38
```

Figure 4: The trace (thin line) of a run after learning line following.

Now, the reflection sense throws away the *absolute*, quantitative values of the reflection sensors and focuses instead on the *difference* between the two sensors to produce the qualitative edge/no edge values. This enables Mr. Mind with the *same* assumptions to follow lines of other colours and even lines where the colour changes along the way. Light seeking and obstacle evasion[1] (also using assumption 0A1 but with other triggering senses), and zigzagging (turn left, go forward, right, forward, left, etc.; i.e. it is necessary to look two time steps back) were also successfully learned in simulation. But for a simulated phase to be meaningful prior to the actual, situated phase, the transfer from simulation to reality should be fairly painless in terms of loss of performance. Here, the idea is to train the robot in the simulator and have it produce some assumptions. Then the ‘*good*’-assumptions would be compiled into rules and downloaded to a LEGO-robot. This was tried with one task: “Go straight until hitting a wall, then turn and go toward the light”. A LEGO robot was built using light and bumper sensors and a LEGO Mindstorms RCX computer[2], and within 2 minutes the robot performed the same task in reality as it had done in simulation; the only parameter that had to be adjusted was the threshold for ‘significant difference’ between the two ambient light sensors.

It is not crucial that the **right**-action of the physical robot corresponds exactly to the **right**-action of the virtual robot. What matters is that the action is an in-place right turn that is appropriately small. Of course, one has to bear in mind that having trained the simulated robot to, e.g., follow a specific line does not necessarily imply that the real robot will be able to follow a physical copy of the simulated line; it just means that the physical robot, having learned the basic line following rules, will be able to follow some class of lines, a class whose defining characteristics are not directly retrievable but are related to for example the timing constants of the actions, the placement of the wheels and so on. However, since for all choices of constants etc. there will always be lines that the robot will be unable to follow, there is no real harm done in making the choices arbitrarily, as long as you are not interested in one particular line only.

III. RELATED WORK

As our assumptions, the classifiers of the learning system ALECSYS [4] are also condition-action rules, each with a “strength” that decides the probability of the rule being executed. However, in the Assumption Architecture the picture is more black and white; contradictory assumptions are not allowed, only one possible explanation for the feedback is upheld. Hence, useless assumptions are discovered quickly and do not delay learning. The price is that the trainer is expected to give predominantly correct feedback.

Another way of looking at the assumptions is to view them as predictors of feedback, composed by predicates over sense values and actions. In [11] an interesting system is described where chains of behaviours are formed by means of such predictors; one crucial point is the ability of stimuli to function also as conditioned reinforcers. This is an idea that we would like to take up in future work to facilitate sequential behaviour. Currently it is not possible for Mr. Mind to learn more than one (sub) task at a time.

Meant for a different and more complex ecological niche, the Hamsterdam toolkit of [1] still bears some resemblance to our system. Blumberg, Todd, and Maes have employed ‘releasing mechanisms’, dedicated filters that monitor the sensors for certain features of the environment. This is what our senses would look like if carried to a higher level of complexity. Further, so-called ‘discovery groups’ are formed so that multiple possible explanations for

[1] You can’t really *avoid* anything with only bumper sensors!

[2] An RCX has a Hitachi H8/300 CPU and 32K RAM, three input and three output ports. www.legomindstorms.com.

reinforcement are checked simultaneously. In our system, as described, only one explanation is checked at a time, and this is upheld as long as it is consistent.

If the trainer and the designer are not necessarily the same person, the flexibility of training versus environment reinforcement [5] is desirable. Our system is general and lets the trainer teach Mr. Mind any simple task expressible as associations between stimuli and action. Our hope is that in providing only the generic assumption types, and not a fixed set of learnable tasks that the trainer essentially chooses from as in [7], more is left to the imagination of the trainer.

IV. CONCLUSIONS

In this work, we have taken as a starting point the *ecological niche* of simplistic LEGO robots and built a learning system using binary training. Acknowledging the principle of ecological balance [9], the tasks considered are also quite simple. In structuring the sensors in pre-processing *senses* interpreting data into information, we achieve decentralisation, abstraction, and fast learning. Also, the senses assisted in an easy transfer from simulation to reality. For such simple tasks that can come into consideration for LEGO robots, appropriate wrappers are typically easy to write. Of course, focusing on some aspects of the available sensor data is ignoring other aspects. A sense is aimed at a certain set of tasks. If the task is not line following but "stay on the blue spot", for example, then the current wrapper will not be very helpful. One therefore might write a pre-determined set of wrappers (and you do not have that many ways of using simple sensors), so that the appropriate senses could be chosen prior to each training session.

Forming assumptions about successful behaviour is a way of structuring experience. Humans might call it abstraction: instead of remembering all the hundreds of individual instances of cats seen so far, one forms the concept of "a cat" based on the similarities of the instances. Once in a while, one might have to refine this concept due to new and atypical instances – but still the concept of "cat" is valuable. Likewise with our robot: some assumptions might prove false, but even so they guide the robot toward more feedback and hence more precise assumptions. Further, we feel that it is a very nice feature to be able to follow the robot's behaviour: with explicit assumptions you can tell why it's doing what it's doing.

In the ecological niche studied here, it is possible to teach a robot various tasks by saying only *good*! and *bad*! to it. This kind of binary training is immediately understandable, also for children. The robot learned each task within 10 minutes, mostly within 3. We adopted a principle of *significance of all feedback*, and this probably has strong influence on the fast learning. Synchronisation of feedback and action made the job easier for the trainer, and so the only training problems are caused by genuine mistakes or by *ethnocentric oversight*, where the trainer forgets that robots do not have the capabilities that our own kind does.

V. FUTURE WORK

The training could take place entirely in reality so that the trainer would interact with the robot via a remote control (as described in [11]) or a speech recognition module. Such situated learning would enhance the notion of an autonomous, live creature that especially children like, but it would also introduce challenges: the synchronisation would be less than trivial to keep with noisy IR-communication, and the robot might have to transmit its sense values for the trainer to see. Further, more assumption types are easily imaginable, e.g. incorporating some form of state.

VI. ACKNOWLEDGEMENTS

Ole Caprani, John Hallam, Henrik Hautop Lund, and our referees made valuable suggestions to this work.

VII. BIBLIOGRAPHY

[1] B. Blumberg, P. Todd, P. Maes, "No Bad Dogs: Ethological Lessons for Learning", Proc. of the Fourth Int. Conf. on Simulation of Adaptive Behavior, Cambridge, Mass. (ed. Maes, Mataric, Meyer, Pollack, Wilson). MIT Press 1996.
[2] V. Braitenberg, "Vehicles, Experiments in Synthetic Psychology", MIT Press, Cambridge, Mass. 1984.
[3] R. Brooks, "A Robust Layered Control System For A Mobile Robot", IEEE Journal of Robotics and Automation, vol. RA-2, No. 1. March 1986.
[4] M. Dorigo, M. Colombetti, "Robot Shaping: Developing Autonomous Agents through Learning", Artificial Intelligence, 71(2):321-370, 1994.
[5] M. Dorigo, M. Colombetti, "The role of the trainer in reinforcement learning"**.** Proc. of MLC-COLT '94 Workshop on Robot Learning, S.Mahadevan et al. (Eds.), July 10th 1994, New Brunswick, NJ, 37-45, 1994.
[6] T. Henderson, E. Shilcrat, "Logical Sensor Systems", Journal of Robotic Systems Vol. 1, No. 2, pp. 169-193, 1984.
[7] H. H. Lund, O. Miglino, L. Pagliarini, A. Billard, A. Ijspeert, "Evolutionary Robotics - A Children's Game", Proc. of IEEE 5th Int. Conf. on Evolutionary Computation. IEEE Press, 1998.
[8] P. H. Mowforth, T. Zrimec, "Learning of Causality by a Robot", Machine Intelligence 12, ed. J. E. Hayes, D. Michie, and E. Tyugu, Clarendon Press, Oxford, 1991.
[9] R. Pfeifer, "Teaching powerful ideas with autonomous robots", Journal of Computer Science Education, 7, 161-186, 1997.
[10] M. Resnick, F. Martin, "Children and Artificial Life", Constructionism (ed. Harel/Papert), p. 379-390. Norwood, NJ: Ablex Publishing 1991.
[11] D. S. Touretzky, L. M. Saksida, "Skinnerbots", Proc. of the Fourth Int. Conf. on Simulation of Adaptive Behavior, Cambridge, Mass. (ed. Maes, Mataric, Meyer, Pollack, Wilson). MIT Press 1996.
[12] S. W. Wilson, "The Animat Path to AI", Proc. of the First Int. Conf. of Simulation af Adaptive Behavior, Cambridge, Mass. (ed. Meyer, Wilson). MIT Press/Bradford Books 1991.